Remarks

Applicants respectfully request reconsideration of the present U.S. Patent application as amended herein. Claim 23 has been amended. No claims have been added or canceled. Thus, claims 1-27 are pending.

CLAIM REJECTIONS – 35 U.S.C. § 103(a)

Claims 1-4, 10-15, 17, 18 and 23 were rejected as being unpatentable over U.S. Patent No. 6,058,491 issued to Bossen, et al. (*Bossen*) in view of U.S. Patent No. 6,023,772 issued to Fleming, et al. (*Fleming*). For at least the reasons set forth below, Applicants submit that claims 1-4, 10-15, 17, 18 and 23 are not rendered obvious by *Bossen* and *Fleming*.

As a preliminary matter, Applicants note that the Office Action fails to address the amendments made in response to the previous Office Action. The Office Action appears to provide a broad brush characterization of Applicants' remarks in order to provide a straw man argument to traverse. See pages 16-17. However, *Bossen* discloses fault detection ***before*** the memory commit point. See col. 3, lines 23-26. In contrast, the claimed invention fault detection ***after*** the memory commit point.

Applicants are not asserting that *Bossen* does not disclose a checkpoint or re-execution of instructions as asserted in the Office Action. Rather, Applicants assert that *Bossen* provides a different scheduling relationship between fault detection and commitment to memory. Further, the claims recite re-executing the non-deterministic memory access events, which is not relevant to *Bossen* because, in *Bossen*, fault detection

has been completed before the results are committed to memory. Therefore, re-executing the non-deterministic memory access events is unnecessary.

Fleming discloses storage of non-deterministic event information as incurring substantial processing overhead and a problem to be overcome. See col. 2, liens 62-65. Further, *Fleming* discloses roll-back as generally not attractive. See. col. 3, lines 3-5. Therefore, *Fleming* clearly ***teaches away*** from the claimed invention.

One of ordinary skill in the art would not have combined *Bossen* with *Fleming* for at least the reasons set forth above. Further, even if *Bossen* and *Fleming* are combined the result does not teach or suggest the claimed invention because neither *Bossen* nor *Fleming* provide fault detection after the memory commit point and re-execute non-deterministic memory access events in response to fault detection. Therefore, *Bossen* and *Fleming* cannot render the claimed invention obvious.

Claim 1 recites:

> storing a register architectural state of a processor corresponding to a first checkpoint;
> storing non-deterministic memory access events as associated data that occur subsequent to the storage of the first checkpoint;
> determining whether an processing error has occurred ***subsequent to the storage of the first checkpoint***; and
> restoring the register architectural state of the processor corresponding to the first checkpoint and re-executing the ***non-deterministic memory access events*** if a processing error is detected.

Similarly, claim 15 recites:

> means for storing a register architectural state of a processor corresponding to a first checkpoint;
> means for storing non-deterministic memory access events as associated data that occur subsequent to the storage of the first checkpoint;
> means for determining whether an processing error has occurred ***subsequent to the storage of the first checkpoint***; and

> means for restoring the register architectural state of the processor corresponding to the first checkpoint and re-executing the ***non-deterministic memory access events*** if a processing error is detected.

Thus, Applicants claim storing a register architectural state corresponding to a checkpoint and logging of non-deterministic memory access events ***subsequent to the checkpoint***. If an error occurs after the checkpoint, the architectural register state is restored from the checkpoint and the ***non-deterministic memory access events*** are re-executed.

Bossen discloses storing the internal state information of one of the processors at each checkpoint. See col. 5, lines 52-55. *Bossen* discloses fault detection before the memory commit point. The processor results of the two processes are compared ***after each macro-instruction***. See col. 7, lines 1-2. When an error is detected, an error-free architectural state is achieved by rolling back the architectural state stored in the register file using the lookback state to obtain the architectural state from two instructions back. See col. 6, lines 38-41; col. 7, lines 19-32 and Figure 6.

1-4, 10-15, 17, 18 and 23

Claims 2-4 and 10-14 depend from claim 1. Claims 17 and 18 depend from claim 15. Because dependent claims include the limitations of the claims from which they depend, Applicants submit that claims 2-4, 10-14, 17 and 18 are not rendered obvious by *Bossen* and *Fleming* for at least the reasons set forth above.

Claim 23 recites:

> leading thread execution circuitry to execute a leading thread of instructions;
> trailing thread execution circuitry to execute a trailing thread of instructions;
> a memory controller coupled with the leading thread execution circuitry; and

> a memory coupled with the leading thread execution circuitry and the trailing thread execution circuitry to store information related to non-deterministic memory access events, wherein the memory controller causes the memory to store a register architectural state of a processor corresponding to a first checkpoint and non-deterministic memory access events as associated data that occur subsequent to the storage of the first checkpoint, the memory controller determines whether an processing error has occurred subsequent to the storage of the first checkpoint, and, in response to the processing error, the memory controller restores the register architectural state of the processor corresponding to the first checkpoint and re-executes the non-deterministic memory access events.

Thus, Applicants claim leading thread execution circuitry, trailing thread execution circuitry where a register architectural state corresponding to a checkpoint and logging of non-deterministic memory access events is stored ***subsequent to the checkpoint***. If an error occurs after the checkpoint, the architectural register state is restored from the checkpoint and the ***non-deterministic memory access events*** are re-executed.

As discussed above, neither *Bossen* nor *Fleming* provide fault detection after the memory commit point and re-execute non-deterministic memory access events in response to fault detection. Therefore, no combination of *Bossen* and *Fleming* can teach or suggest the invention as claimed in claim 23.

ALLOWABLE SUBJECT MATTER

Claims 5-9, 16, 19-22 and 24-27 were objected to for being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. In light of the discussion above, Applicants submit that claims 5-9, 16, 19-22 and 24-27 are in condition for allowance.

CONCLUSION

For at least the foregoing reasons, Applicants submit that the rejections have been overcome. Therefore, claims 1-27 are in condition for allowance and such action is earnestly solicited. The Examiner is respectfully requested to contact the undersigned by telephone if such contact would further the examination of the present application. Please charge any shortages and credit any overcharges to our Deposit Account number 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Date: November 14, 2007

/Paul A. Mendonsa/
Paul A. Mendonsa
Attorney for Applicant
Reg. No. 42,879

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(503) 439-8778